資源制約付きプロジェクト・スケジューリング問題に関する基礎的研究 (21世紀の数理計画 : 最適化モデルとアルゴリズム)

AUTHOR(S):

藤原, 稔久; 諏訪, 晴彦; 森田, 浩

CITATION:

藤原, 稔久 ...[et al]. 資源制約付きプロジェクト・スケジューリング問題に関する基礎的研究 (21世紀の数理計画 : 最適化モデルとアルゴリズム). 数理解析研究所講究録 2009, 1629: 125-130

ISSUE DATE:

2009-02

URL:

http://hdl.handle.net/2433/140362

数理解析研究所講究録
第1629巻 2009年 125-130

資源制約付きプロジェクト・スケジューリング問題に関する基礎的研究

大阪大学大学院・情報科学研究科　藤原　稔久 (Toshihisa Fujiwara)
摂南大学・工学部　諏訪　晴彦 (Haruhiko Suwa)
大阪大学大学院・情報科学研究科　森田　浩 (Hiroshi Morita)

Graduate School of Informatics Science
Osaka University
Dept. of Industrial and Systems Engineering
Setsunan University
Graduate School of Informatics Science
Osaka University

概要

A predictive project schedule in project management undergoes various kinds of contingencies such as activity delays and lack of resources during project execution. Under such a dynamic environment, a reactive scheduling strategy will be effective to cope with uncertainties to keep feasibility and the quolity of the schedule. This paper deals with the reactive scheduling strategy for resource constrained project scheduling problems, and investigates the typical reactive approaches from the viewpoint of when-to-schedule policy and incremental costs for resources.

1　はじめに

プロジェクト管理においては，高品質なプロジェクト・スケジュールを立案するだけでなく，プロジェクトの実行段階で発生する，不確定的事象にうまく対処しつつ，スケジュールの実行可能性を維持していくことが肝要である [1]．このような動的環境下でのスケジューリング意思決定の枠組みを動的スケジューリングと呼ぶ．とくに，高品質なスケジュールを維持することを目的としてスケジュール修正を実施する戦略は，リアクティブ戦略と呼ばれている．現実のプロジェクト管理では，スケジュールの遅延は必ず生じるものである，という楽観的な前提の下，意思決定者の経験や勘に基づいて資源の再配置やスケジュール修正を実施する場合が少なくない [2]．さらに，プロジェクト・スケジューリングにおけるリアクティブ戦略の意思決定モデルは確立しておらず，ヒトの意思決定を主体とした現実のリアクティブ戦略の妥当性・有効性に対する客観的評価はなされていないのが実情である．

一般に，リアクティブ戦略といえば，事象駆動型 (event-driven) 方策 [4]，あるいは定期型 (periodic) 方策 [6] を指すことが多い．事象駆動型方策は即応性の観点からは妥当であると考えられるが，過度のスケジュール修正を招いたり，対象プロジェクトが常にスケジューリング状態下に置かれるため，必ずしも効果的なスケジュール変更を実施しているとはいえない．これに対して諏訪らは，生産スケジューリングにおいて，不確定的事象の情報を集約する形で累積遅延の概念を導入し，累積遅延に基づくスジューリング方策（以下，D^*p）を提案している [5]．

本研究では，プロジェクト・マネジメントの一環として，プロジェクト・スケジューリングへのこれらのスケジューリング方策の導入を考える．ここでは，資源制約付きプロジェクト・スケ

ジューリング問題（resource constrained project shceduling problem；RCPSP）において上述の三つのリアクティブ戦略の実行性能を比較検討する．

2 プロジェクトにおけるリアクティブ戦略の概要

2.1 記号と仮定

プロジェクトの計画期間を $[0,T]$ とする．区間 $[0,T]$ 上の時点 $\tau_i = i\tau$ $(i = 0,1,\cdots,M, \tau > 0)$ を考える．ただし，$\tau_0 = 0$ はスケジュール実行の開始時点とする．ここで，次の記号を定義する．

J_i^{A}: 時点 τ_i で処理が完了している作業の集合
J_i^{P}: 時点 τ_i で未処理の作業の集合
S_i^{A}: 時点 τ_i での J_i^{A} に対する実際のスケジュール
S_i^{P}: J_i^{P} に対する τ_i 以降の予定のスケジュール

各時点 τ_i $(i > 0)$ において，以下に示す点検とスケジューリングを実施するとする．以下では，スケジューリングとは区間 $(0,T]$ で実施するスケジュールの変更（再スケジューリングや部分スケジュールの修正）を表す．点検はプロジェクトの進捗状況を把握するためのモニタリングを指し，離散時間間隔 τ $(= \tau_i - \tau_{i-1})$ で行うものとする．モニタリングの対象は不確定的事象やそれにともなうプロジェクト遅延の生起時刻と時間的規模などがある．点検時点において，スケジューリングを実施するかどうかの判断が必要であれば，その意思決定を行うとする．

一方，スケジューリングは τ_i 以降に割り付けられている未処理の作業を対象とする．すなわち，時点 τ_i におけるスケジューリングとは所与の目的関数（メイクスパン）の最小化を目的として，J_i^{P} に対するスケジュール S_i^{P} を生成することである．このようなスケジューリング問題は，RCPSP の枠組みで扱われるとする．これら，点検およびスケジューリングを基準にすれば，プロジェクトにおけるリアクティブ戦略は次のようにまとめることができる．

1. 定期型
   点検間隔 $\tau = T/M$ とし，各時点 τ_i $(= iT/M)$ でスケジューリングを実施する．すなわち，点検間隔とスケジューリングの実施間隔が等しく，この定期型方策の設計変数はスケジューリングの実施回数を表す M である．

2. 事象駆動型
   予め定められた点検間隔 τ でスケジュールの進捗状況を見る．現時点 τ_i において，区間 $(\tau_{i-1}, \tau_i]$ に突発的な事象（資源変動や新規作業の発生/キャンセル）が生起した場合，この時点でスケジューリングの実施の判断を行う．実施の判断をしたとすると，τ_i（ないしは τ_i 以降のある点検時点）以降に割り付けられている作業を対象として，スケジューリングを実施する．事象駆動型での設計変数は τ であるが，後述するようにスケジューリングの契機となる事象をどう扱うかが重要となる．他方，実施しない場合は，τ_i 以降のスケジュールに対して右シフト（right-shift）を適用する．事象駆動型の方策では，スケジューリング実施の判断基準をどのように設定するかが肝要となる．

3. 累積遅延型
   スケジュール区間 $[0,T]$ 上の時点 $\tau_i = i\tau$ $(i = 0,1,\cdots, m, \tau > 0, \tau_m \leq H)$ において，以

下に述べる点検と，場合に応じてスケジューリングを実施するものとする．点検とはスケジュールの進捗状況を把握するためのモニタリングを指す．点検の対象は作業の遅延とし，点検時点 τ_i において，作業の遅延個数，未処理の作業数，各作業の遅延時間を計測する．未処理の作業を対象としたスケジューリング問題は，一般の静的スケジューリングの枠組みで扱うこととする．特に $i \geq 1$ なる時点 τ_i において，未処理の作業すべてを対象とするスケジューリングを再スケジューリングと呼ぶ．

定期型のアプローチはローリング・スケジュール（rolling schedules）に代表されるように，古くから研究が行われている．タイミングの決定は比較的容易に行えるものの，環境変動に対して必ずしも迅速に対応できるとは限らない．事象駆動型のアプローチは知識ベース・スケジューリング・システムで頻繁に採用されている方式であり，環境変動に迅速な対応を実現している．しかしながら，大規模システムにおいて予測困難な事象が頻繁に起こる場合，対象システムが常にスケジューリングの状況下に置かれるため，スケジューリングにかかる計算コストが増大する恐れがある．また，事象を契機として意思決定を行うため，影響度合いに応じた事象のクラス分けが必要となる．次に，累積遅延型のアプローチはスケジューリング意思決定の判断基準として累積遅延時間（以下，累積遅延）を定義し，制御限界方策のアイディアに基づき累積遅延が限界値（限界累積遅延 D^*）を越えていた場合にスケジュール修正が実施される．累積遅延型では，上記の制御限界方策のアイディアに基づいたリアクティブなスケジュール修正を実現することにより，事象のクラス分けの問題を解消している．

3 対象問題の記述

本研究では，以下に述べる単一プロジェクトに対するRCPSPを対象とする．作業集合 $\mathcal{J}$，および各時間ごとの使用可能量の上限を持つ n 種類の資源（$R_1, R_2, \cdots, R_n$）が与えられている．作業とは，各資源について一定の資源量を使用することによって処理されるものとする．資源は作業ごとに決められた作業時間の間は作業によって使用されるが，作業完了後は再び使用可能となることとする．作業ごとに与えられた先行順序関係を満たした上で，最終作業の作業完了時刻を最小化するように資源を作業に割り当て，作業の開始時間を決定する．

3.1 基本モデル

基本モデルの定式化で用いる集合および入力データを示す．

集合

- $\mathcal{J}$：作業集合を $\mathcal{J} = \{1, 2, \cdots, N\}$ とする．
- $\mathcal{T}$：時刻を表す集合を $\mathcal{T} = \{1, 2, \cdots, T\}$ とし，時刻 $(t-1)$ から時刻 t までを区間 t とする $(1 \leq t \leq T)$．
- $\mathcal{R}$：資源集合を $\mathcal{R} = \{R_1, R_2, \cdots, R_n\}$ とする．
- $\mathcal{P}$：作業の先行関係の集合とする．$(j,k) \in \mathcal{P}$ のとき，作業 j の処理が終了するまで作業 k の処理が開始できないことを意味する．ただし作業 j, k $(1, 2, \cdots N)$ とする．

入力データ

- p_j：作業 $j(j=1,2,\cdots,N)$ の処理時間
- c_{jt}：作業 j を時刻 t から開始したときの資源量
- R_{rjt}：区間 t での作業 j の処理に要する資源 $r(r=1,2,\cdots,n)$ の量
- RU_{rt}：区間 t における資源 r の利用可能量

変数

- x_{jt}：作業 j を時刻 t に開始するとき 1，それ以外は 0 を表す．

基本モデルの定式化

$$\text{min.} \quad \sum_{j\in\mathcal{J}} \sum_{t=1}^{T-p_j+1} c_{jt}x_{jt} \tag{1}$$

$$\text{sub.to} \quad \sum_{t=1}^{T-p_j+1} x_{jt} = 1 \quad \forall j \in \mathcal{J} \tag{2}$$

$$\sum_{j\in\mathcal{J}} R_{rjt} \sum_{s=\max\{t-p_j+1,1\}}^{\min\{t,T-p_j+1\}} x_{js} \le RU_{rt} \tag{3}$$

$$\forall r \in \mathcal{R},\ \ t \in \mathcal{T}$$

$$\sum_{t=2}^{T-p_j+1} (t-1)x_{jt} + p_j \le \sum_{t=2}^{T-p_j+1} (t-1)x_{kt} \tag{4}$$

$$\forall (j,k) \in \mathcal{P}$$

$$x_{jt} \in \{0,1\} \quad \forall j \in \mathcal{J}, t = 1,\cdots,T-p_j+1 \tag{5}$$

上の (2) 式は，すべての作業は必ず一度処理されなければならないことを表す．(3) 式は，ある時刻 t に処理中である作業の資源使用量の合計が，資源使用量の上限を超えてはならないことを示す．また，時刻 t に処理中である作業は，時刻 $(t-p_j+1)$ から t の間に処理を開始した作業に対応する．(4) 式は，作業 j の処理が終了するまで作業 j が先行作業となる作業 k の処理が開始できないことを表す．左辺は，作業 j の完了時刻を表し，右辺は作業 k の開始時刻の直前の時刻を表す．

本研究では，所与のプロジェクト・スケジュールの実行過程において，不確定的事象がランダムに発生するものとする．ここでいう不確定的事象とは，プロジェクトのメイクスパンに増加を生じさせる事象を指す．

4 数値実験

数値実験により，2章で述べた，三つの方策の有用性の比較検討をする．また，不確定的事象発生の影響により先行制約，資源制約を違反する場合は，資源を追加することによりこれを回避できることとし，これにかかる費用を資源追加コストと呼ぶ．

4.1 実験条件

数値実験には，RCPSPのベンチマーク問題であるPSPLIB[3]の$j60.\mathrm{sm}(n=60)$，$j90.\mathrm{sm}(n=90)$を用いた．(2),(3),(4)および(5)式を満たす初期スケジュールを，局所探索法により生成する．プロジェクト実行時に発生する不確定的事象については，作業時間と資源量の変動（減少）を取り上げる．作業時間の増加のイベントは，時間軸上で平均$1/\lambda$の指数分布に従って発生するものとし，その対象作業の作業時間の増加量は平均$1/\mu$の指数分布に従うものとする．次に，資源量の減少は，時間軸上で平均$1/\gamma$の指数分布の従って発生する．このときの資源減少の時間区間は，平均$1/\sigma$の指数分布に従うものとする．数値実験では，$\lambda=0.1,0.2,\mu=1,\gamma=0.1,\sigma=0.25$とし，資源の減少量は1とする．$D^*p$方策については，限界累積遅延$D^*=10,15$，点検間隔$\tau=5,10$の計2種類を考えた．スケジュール修正方法については，スケジュール修正時点以降の作業を対象とし先行制約を遵守したシフトスケジューリングを採用する．

4.2 比較実験

4.1で述べた2種類の不確定的事象の発生シナリオについて各方策におけるスケジュール修正のシミュレーションを500回行った．表1および表2に各問題例と不確定的事象シナリオの組合せに対する結果をまとめておく．実験結果では，各方策に対するプロジェクト実行結果としてのスケジュールの修正頻度（freq.），全資源追加コスト（cost），およびメイクスパン（makespan）のそれぞれの平均を示している．表1から，D^*p方策は，事象駆動型と定期型方策に比べて，スケジュール修正頻度を抑えつつ資源追加コストおよびメイクスパンの値が低いことがわかる．このことから，D^*p方策は他の方策と比べ有用であるといえる．不確定的事象の発生間隔とD^*p方策の関係に着目すると，頻繁に不確定的事象が発生する場合においてはD^*p方策が設定変数値に関わらずスケジュール修正頻度の大幅な低減させていることがわかる．さらに，資源追加コストとメイクスパンのそれぞれの平均値においても良好な結果を示している．このことは，スケジュール計画段階での予定メイクスパンを劣化させることなくスケジュール修正頻度を抑えているといえる．以上のことからD^*p方策は，定期型，事象駆動型方策に比べ有用であることがわかる．

5 おわりに

本研究では，RCPSPへのリアクティブ戦略の有用性について検討した．実験の結果，D^*p方策が，定期型，事象駆動型方策と比べ，スケジュール修正頻度のみならず資源追加コストおよびメイクスパンを抑えることが可能であることがわかった．今後，資源の種類やマルチモード状況下での検討が必要である．また，作業時間増加，資源量の変動のそれぞれがメイクスパンにどのような影響を与えるかという問題も検討する必要がある．

表 1: 実験結果

$\lambda = 0.1$

$n = 60$

	D^*p	periodic	event-driven
freq.	5.2	10.0	59.0
cost	621.8	1113.1	1262.6
makespan	138.6	109.6	96.2

$n = 90$

	D^*p	periodic	event-driven
freq.	4.4	9.1	59.0
cost	1136.0	2157.7	2371.5
makespan	152.3	100.0	428.8

表 2: 実験結果

$\lambda = 0.2$

$n = 60$

	D^*p	periodic	event-driven
freq.	10.5	12.5	89.0
cost	1051.0	1428.7	1606.4
makespan	157.4	138.9	119.3

$n = 90$

	D^*p	periodic	event-driven
freq.	8.3	11.2	89.0
cost	2203.3	1768.4	2992.3
makespan	153.3	123.0	106.2

参考文献

[1] Peter Brucker, Complex Scheduling, Springer, 2006

[2] T.Demarco,Slack:getting past burnout,buswork and the myth of total efficiency, Broadway Books, 2002

[3] R.Kolisch and A.Sprecher : PSPLIB - A Project Scheduling Problem Library ; *European Journal of Operational Research* Vol.96 , No.1 , pp.205 - 216(1997)

[4] I.Sabuncuoglu and O.B.Kizilisik : Reactive Schedulng in a Dynamic and Stochastic FMS Environment; International Journal of Computer Integrated Manufacturing ; Vol.41 , No.17 , pp4211 - 4231(2003)

[5] 諏訪・三道, リアクティブ・スケジューリングにおけるスケジュール修正時期の判断基準に関する一考察 ; システム制御情報学会論文誌, Vol.15 , No.7 , pp327 - pp335(2002)

[6] I. M. Ovacik and R. Uzsoy ; Decomposition Methods for Complex Factory Scheduling Problems ; Kluwer Academic Publishers (1997)